



Sami Anglé

PELINKEHITYS

Demosta julkaisuun

PELINKEHITYS

Demosta julkaisuun

Sami Anglé
Opinnäytetyö
syksy 2014
Tietojenkäsittelyn
koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma, Websovellus kehitys

Tekijä(t): Sami Anglé
Opinnäytetyön nimi: Pelinkehitys demosta julkaisuun
Työn ohjaaja: Viitala Matti
Työn valmistumislukukausi- ja vuosi: Syksy 2014 Sivumäärä: 28

Peliteollisuudesta on jatkuvasti tulossa suurempi ala. Samalla tulee koko ajan enemmän alustoja, jonne pelejä julkaistaan. Osaa peleistä ei koskaan julkaista, koska yrityksillä ei ole varaa kehittää peliä loppuun asti tai kehitysvaiheessa, että peli ei ole sopiva pelaajille.

Tämän opinnäytetyön tarkoituksena oli jatkokehittää Oulu Game Labissa aloitettua peli-demoa ja valmistaa se julkaisuun Google Play -kaupassa. Peliä kehitettiin alun perin Facebookiin, mutta myöhemmin ilmenneiden ongelmien takia pelistä päätettiin tehdä yksinkertaisempi versio Android-alustalle.

Opinnäytetyö-projektissa käytetään Unity-pelinkehitystyökalua, jolla myös demo oli kehitetty. Teoriaosuudessa kerrotaan pelikehityksen suunnittelusta, ohjelmointikielistä ja julkaisualustoista. Lisäksi käydään läpi sitä millaisia ongelmia ohjelmoinnissa voi tulla vastaan ja miten ohjelmasta voi tehdä mahdollisimman toimivan, helposti päivitettävän ja koodista mahdollisimman uudelleen-käytettävän.

Projektin lopputuloksena Hamsterscape Weekly Challenge -peli julkaistiin Google Play-kaupassa. Kehitystyötä voi jatkaa kehittämällä samasta pelistä versiot Windows Phone- ja IOS alustoille. Tulevaisuudessa peliin suunnitellaan viikoittain vaihtuva kenttä, jolloin pelaajilla pysyy mielenkiinto peliin.

Asiasanat: pelinkehitys, demo, julkaisu, Unity, ohjelmointi

ABSTRACT

Oulu University of Applied Sciences
Business Information Systems, Option of web application development

Author(s): Sami Anglé

Title of thesis: Game development from demo to release

Supervisor(s): Viitala Matti

Term and year when the thesis was submitted: autumn 2014

Number of pages: 28

Gaming industry is growing constantly. At the same time the number of publishing platforms is increasing. Some games are never published because companies can not finalize game development. They might also realize that the game is not suitable for gamers.

The purpose of this thesis was to further develop a game demo, made in Oulu Game Lab, and prepare it for release in Google Play store. The game was originally created for Facebook, but due to the problems raised during development, it was simplified and transferred to the Android platform.

In this thesis, the project uses Unity game development tool, which was also used in the development of the demo. The theoretical background describes development of game design, programming languages and publishing platforms. The report also studies different kinds of programming problems which may be encountered and describes how the program can be made as effective as possible, easily updated. Goal was also make programming code reusable as possible

As a result of the project Hamsterscape Weekly Challenge game was released in Google Play- store. Development can be continued by porting the game to Windows Phone and iOS platforms. In the future, the game is planned to be updated with a weekly changing level, that will keep the game interesting.

Keywords: game development, demo, release, Unity, programming

Sisällys

1 JOHDANTO.....	6
2 KEHITYSALUSTA - UNITY.....	7
3 PELIALUSTAT.....	9
3.1 Facebook.....	9
3.2 Android.....	10
3.3 iOS.....	10
3.4 Windows Phone.....	11
4 UNITYN TUKEMAT OHJELMOINTIKIELET.....	12
4.1 C#.....	12
4.2 Javascript.....	13
4.3 Boo.....	14
5 HAMSTERSCAPE - TAUSTAA.....	15
5.1 Alkuperäinen idea.....	15
5.2 Oulu Game Lab ja suunnittelu.....	16
5.3 Markkinointi.....	16
6 TESTAUS.....	17
6.1 Demo-versio.....	17
6.2 Julkaisu-versio.....	17
6.3 Hyödyt.....	18
7 HAMSTERSCAPE-DEMON KEHITYSTYÖ.....	20
7.1 Vaatimukset.....	20
7.2 Aikataulu.....	20
8 HAMSTERSCAPE-JULKAISUVERSION KEHITYSTYÖ.....	22
8.1 Vaatimukset.....	23
8.2 Ohjelmointi.....	23
9 POHDINTA.....	25
10 LÄHTEET.....	26

1 JOHDANTO

Opinnäytetyössä tehtyä Hamsterscape peliä tehtiin alun perin Oulu Game Labissa, jossa peli tehtiin demoon asti. Peli oli aluksi Facebook-peli. Myöhemmin Hamsterscapesta päätettiin tehdä Androidille pienempi versio pelistä ja se sai nimeksi Hamsterscape Weekly Challenge.

Kiinnostus peliohjelmointiin alkoi, kun pääsin harjoittelemaan Oulu Game Labiin, jossa koulutetaan pelialalle ihmisiä. Oulu Game Labissa pääsin ryhmään, joka alkoi lopulta tekemään Hamsterscapea. Harjoittelun jälkeen ryhmässä löytyi kiinnostuneita tekemään peliä eteenpäin ja näin ollen halusin tehdä opinnäytetyön Hamsterscape pelistä.

Opinnäytetyössä oli aluksi tarkoitus tehdä Facebookiin peli, mutta myöhemmin suunnitelmat muuttuivat ja peli tehtiin Android-mobiilikäyttöjärjestelmälle. Hamsterscapesta saatiin tehtyä Androidilla toimiva versio, mutta sitä ei koskaan julkaistu missään. Hamsterscape Weekly Challenge tehtiin Hamsterscapen pohjalta ja se suunniteltiin olemaan mobiilipeli.

Pelien kehitys mobiilialustalle eroaa esimerkiksi Facebookiin tekemisestä suurimmaksi osaksi kosketusnäytön ja näyttöjen tarkkuuksien osalta. Facebookissa oli erittäin rajallinen tila, johon peli pitää mahduttaa. Mobiililaitteilla taas laitteiden eri resoluutiot ja laite tehot vaihtelevat todella paljon. Tämän vuoksi peli pitää suunnitella toimimaan eri kuvasuhteille ja eri laitteistotehoille.

2 KEHITYSALUSTA - UNITY

Unity on Unity Technologiesin kehittämä pelinkehitystyökalu, jolla voi kehittää pelejä alustoille PC, Mac, Linux työpöytä, WEB, iOS, Android, Windows Phone 8, Windows Store, BlackBerry 10, Wii U, Sony PS3 ja -PS4 ja Xbox One (Unity Technologies 2014a, viitattu 10.8.2014). Unityn kehitys aloitettiin vuonna 2001 ja ensimmäinen Unity1 julkaistiin Applen WWDC:ssä vuonna 2005. Tästä Unityn kehittäminen on jatkunut aina tähän päivään asti ja siitä on tullut todella suosittu kehitystyökalu pelinkehittäjien keskuudessa. Vuonna 2010 julkaistiin Unity3:n kanssa samaan aikaan Asset Store, jossa kehittäjät voivat joko jakaa tai myydä omia ratkaisujaan muille kehittäjille. (Unity Technologies 2014b, viitattu 10.8.2014.)

Unity käyttää ohjelmointiympäristönä Monoa, jonka kanssa voi käyttää ohjelmointikieliä JavaScript, C# tai Boo. Kaikkia ohjelmointikieliä voi käyttää Mono ohjelmointiympäristössä. Mono auttaa myös Unityn kanssa ohjelman virheiden jäljittämisen ja mahdollistaa koodin läpikäymisen rivi riviltä. (Unity Technologies 2014c, viitattu 10.8.2014.)

Unityssä on mahdollista käyttää myös Asset Storea, jossa on muiden kehittäjien koodiratkaisuja, tekstuureja, 3d-malleja, animaatioita, jopa kokonaisia esimerkkiprojekteja ja tutoriaaleja. Asset Storesta pystyy lataamaan assetteja, jotka ovat muiden käyttäjien tekemää valmista koodia, grafiikkaa tai muita ratkaisuja, ja tuomaan ne oman projektin käyttöön (Unity Technologies 2014e, viitattu 13.9.2014). Asset Store tekee asioita helpommaksi, mutta usein asset-tiedostoja käytettäessä se täytyy muokata omaan käyttöön sopivaksi. Muokkauksen takia asset-tiedostoon tuleva päivitys voi rikkoa assetin tai tehdä koko pelistä toimimattoman, kun käyttäjä on muokannut assetissa olevaa koodia ja silloin koodin päivityksessä koodin tulee virheitä.

Asset Storeen täytyy tehdä tunnukset, jonka jälkeen sieltä voi ladata ja ostaa vapaasti siellä olevia ratkaisuja omaan käyttöön (Unity Technologies 2014f, viitattu 13.9.2014). Kaupallisia tuotteita tehdessä pitää muistaa tarkistaa asset-tiedoston käyttöön liittyvät käyttöoikeudet, jotta säästytään mahdollisilta yllätyksiltä.

Jos käyttäjällä on Unity päällä ja hän käynnistää Asset Storen, osaa Unity tuoda ladattavat assetit suoraan auki olevaan projektiin. Unity tekee automaattisesti assetin mukana olevat kansiorakenteet, jolloin assetin saa käyttöön suoraan ja Asset Storesta voi myös katsoa, onko jo valmiiksi ladattuihin/ostettuihin asetteihin päivityksiä. (Unity Technologies 2014f, viitattu 13.9.2014.)

3 PELIALUSTAT

HamsterScapesta piti alun perin tulla Facebook-peli, mutta myöhemmin projektin muuttuessa päätettiin peli julkaista Androidille ja pelin nimeksi muutettiin Hamsterscape Weekly Challenge. Näin peli tavoittaa enemmän ihmisiä ja saa paremmin näkyvyyttä. Windows Phone -käyttöjärjestelmälle peliä ei lopulta julkaistu, koska Unity Asset-storesta käyttämämme asset-scripti ei toiminut Windows Phone -käyttöjärjestelmässä oikein.

3.1 Facebook

Facebookiin tulee jatkuvasti uusia pelejä, joten on hyvä tietää, millaisia pelejä Facebook sisältää ja millaisia ovat normaalit Facebook-pelien pelaajat. Samalla myös on hyvä muistaa, mikä Facebook on ja miten sitä voidaan käyttää hyödyksi pelejä kehittäessä.

Nykyään Facebookia käytetään moniin tarkoituksiin. Facebookin avulla ollaan yhteyksissä ystäviin, laitetaan kuvia nettiin ja vaikka pidetään blockia tai pelataan pelejä. Siellä jokaiselle rekisteröityneelle annetaan oma profiili, jota voi muokata ja tämän avulla kaverit löytävät käyttäjän. (Kajaanin ammattikorkeakoulun kirjasto, viitattu 10.8.2014.)

Facebook-alusta salliin Facebookin ulkopuolisten kehittäjien tehdä Facebookiin omia sovelluksiaan, kuten pelejä ja ohjelmia. Sovellukset käyttävät Facebookissa käyttäjien tietoja ja antavat näin kehittäjälle mahdollisuuden tehdä sosiaalisia ominaisuuksia sovelluksiinsa. (Catone. 2009 ,viitattu 10.8.2014.)

3.2 Android

Alun perin Androidia kehitti Android Inc. Android perustuu Linuxin avoimen lähdekoodin käyttöjärjestelmään ja se on kehitetty kosketusnäyttölaitteille. Google osti Androidin vuonna 2003 (Wikimedia Foundation, Inc 2014b, viitattu 29.9.2014). Android käyttöjärjestelmää käytettiin 45%:ssa vuonna 2014 myydyissä mobiililaitteissa (NetMarketShare 2014, viitattu 29.9.2014).

Android-alustalle HamsterScapea kehittäessä ei ollut suuria ongelmia, koska Unityssä monet komennot, jotka toimivat web-sovellusta tehtäessä, toimivat myös kosketusnäytössä. Kosketusnäytölle on tarpeellista tehdä oma koodi, jos kosketusnäytössä tarvitsi kosketusnäytön swipe-toimintoa eli sitä kun sormea liikutetaan kosketusnäytössä. Tätä ominaisuutta käytettiin, kun kehitettiin HamsterScapen Facebook-versiota Androidille. Tehtäessä pelistä käännöstä Androidille tarvitsee ohjelmoijan ladata Android SDK, jonka avulla hän saa tehtyä pelistä apk-tiedoston, josta pelin pystyy asentamaan Android-laitteelle. Hamsterscape Weekly Challenge julkaistiin vain Androidille.

3.3 iOS

iOS on Applen kehittämä käyttöjärjestelmä kosketusnäytöllisille mobiililaitteille ja se on käytössä kaikissa Applen älypuhelimissa ja kämmentietokoneissa. iOS-käyttöjärjestelmällä ei aluksi ollut virallista nimeä, joten käyttöjärjestelmää kutsuttiin nimellä OSX. Vuonna 2008 iPhone SDK:n testiversion julkaisun jälkeen käyttöjärjestelmän nimi muutettiin iPhone OS:ksi ja lopulta vuonna 2010 Apple muutti nimen vuotuisessa WWDC-konferenssissa iOS:ksi. (Wikimedia Foundation, Inc 2014c, viitattu 2.10.2014.) iOS on nykyään lähes yhtä suosittu käyttöjärjestelmä puhelimissa kuin Android. iOS:ää käytettiin 44%:ssa myydyistä mobiililaitteissa vuonna 2014 (NetMarketShare 2014, viitattu 29.9.2014).

Ohjelmoijan kannalta Applen iOS-käyttöjärjestelmälle tehty pelikehitys Unityllä eroaa eniten pelin käännöksen tekemisessä iOS-käyttöjärjestelmälle. Unity tekee pelistä Xcode:lla avattavan projektin, jolla pelistä saa iOS:llä ajettavan ohjelman. Ongelmaksi muodostuu se, että Xcodea ei saa asennettua kuin Applen laitteille.

3.4 Windows Phone

Windows Phone on Microsoftin kehittämä käyttöjärjestelmä, joka on tarkoitettu mobiililaitteille. Windows Phone tuli jakeluun vuonna 2010 ja Windows Phone -käyttöjärjestelmää käyttäviä puhelimia valmistavat Nokia, Samsung ja HTC. Microsoftin ostaessa matkapuhelinliiketoiminnan Nokialta Microsoft pystyy nyt hallitsemaan Windows Phone -puhelinten myyntiä itse. Käyttöjärjestelmä perustuu Windows 8 -käyttöjärjestelmässäkin olevaan Metro käyttöliittymään. (Wikimedia Foundation, Inc 2014d, viitattu 2.10.2014.)

Unityllä ohjelmoitaessa usein koodit toimii yhtä hyvin Windows Phone kuin Android käyttöjärjestelmässä. Windows Phonelle kehittäessä ohjelmoijan tarvitsee asentaa Visual Studio -ohjelma, jossa on Windows Phone SDK mukana. Unity tekee projektista Visua Studio-ohjelmalla avattavan projektin. Visual Studiolla voidaan asentaa peli suoraan puhelimeen.

4 UNITYN TUKEMAT OHJELMOINTIKIELET

Ohjelmointikielen valinnassa vaikeuksia tuottaa usein tietoisuus siitä että ohjelmointikielillä on omat vahvuutensa. Ohjelmointikielen valintaan vaikuttaa myös ohjelmoijan omat mieltymykset. Internetistä löytyy hyvin apua ohjelmointiin. Merkittävin ero tulee siitä miten Unity Asset Storesta käytettäviä Asset-tiedostoja onnistuu käyttämään milläkin ohjelmointikielellä ja millaisia ovat eri ohjelmointikielten syntaksit.

4.1 C#

C# (C sharp) on Microsoftin vuonna 1999 kehittämä ohjelmointikieli, jonka kehittäjänä on Andres Hejlsberg. Alussa kieli tunnettiin nimellä COOL(C-like Object Oriented Language). Nimi muutettiin vuonna 2000 C# Professional Developers Confrensissa. C# kehityksen tavoitteina oli kehittää yksinkertainen, uudenaikainen, yleisiin tarpeisiin sopiva olio-ohjelmointikieli. (Wikimedia Foundation, Inc 2014a, viitattu 17.9.2014.)

Internetistä haettavat dokumentaatiot löytyvät usein paremmin C#:lle (Catlike Coding 2014, viitattu 16.8.2014). HamsterScapea lähdettiin alun perin ohjelmoimaan JavaScript ohjelmointikielellä, mutta Unityn assett-scriptien yhteensopivuusongelmien takia ohjelmointikieli vaihdettiin lopulta C#:iin.

```
using UnityEngine;

using System.Collections;

public class ExampleSyntax : MonoBehaviour
{
```

```

int myInt = 5;

int MyFunction (int number)
{
    int ret = myInt * number;
    return ret;
}

```

KUVIO 1. C#-esimerkki koodi (Unity Technologies 2014d, viitattu 16.8.2014.)

4.2 Javascript

Monet ihmiset kutsuvat Unityssä käytettävää UnityScriptiä JavaScriptiksi, jopa Unity itse kutsuu sitä JavaScriptiksi. Kun Unityllä ohjelmoitaessa tarvitsee hakea apua internetistä, hakeminen vaikeutuu, koska vastauksina tulee usein JavaScriptin koodia UnityScriptin tilalla. (wiki.unity3d.com 2014, viitattu 13.9.2014.)

UnityScriptillä ohjelmoitaessa on eroja JavaScriptiin. UnityScriptissä muuttujille määritellään heti muuttujan tietotyyppi. JavaScript-ohjelmointikielellä muuttujat määritellään vasta silloin, kun muuttujalle annetaan joku arvo. UnityScriptissä on myös luokat, jotka tekevät koodin luettavammaksi kuin JavaScript. (wiki.unity3d.com 2014, viitattu 13.9.2014.)

```
#pragma strict
```

```
var myInt : int = 5;
```

```
function MyFunction (number : int) : int
```

```
{
    var ret = myInt * number;

    return ret;
}
```

KUVIO 2. JavaScript esimerkki koodi (Unity Technologies 2014d, viitattu 16.8.2014.)

4.3 Boo

Ohjelmointikielenä Boo muistuttaa syntaksiensa osalta Pythonia ja se on vähiten suosittu ohjelmointikieli Unityn kolmesta vaihtoehdosta. Boota ei suositella aloittelijalle, koska Boo on Python perusteinen, jossa käytetään duck typing ohjelmointityyliä (Pleasant-Ryan 2012, viitattu 16.8.2014). Duck typing tarkoittaa ohjelmointityyliä jossa muuttuja tehdään määrittelemättömäksi, jolloin muuttuja voi olla mitä tyyppiä vain (The Codehaus 2006, viitattu 16.8.2014).

```
import UnityEngine
import System.Collections

public class ExampleSyntax(MonoBehaviour):

    private myInt = 5

    private def MyFunction(number as int) as int:
        ret as int = (myInt * number)

        return ret
```

KUVIO 3. Boo esimerkki koodi (Unity Technologies 2014d, viitattu 16.8.2014)

5 HAMSTERSCAPE - TAUSTAA

HamsterScapea ryhdyttiin suunnittelemaan Oulu Game Labissa. Idean päästyä läpi demo-vaiheeseen HamsterScape äänestettiin 2013 OGL Expossa parhaaksi esitetyksi demoksi. HamsterScapesta oli tarkoitus tehdä älypeli Facebookiin, jossa palikoita pelikentällä siirtelemällä saadaan tehtyä reitti maaliin.

Pelin suunnitelmat muuttuivat demon jälkeen, kun pelille tarvittiin rahoitusta ja ei ollut aikaisempia näyttöjä jotta rahoittajat kiinnostuisivat. Pelistä päätettiin tehdä pienempi versio ja julkaista se Google Play -kaupassa.

5.1 Alkuperäinen idea

Suunnitteluvaiheen alussa pelin nimi oli Glasses-lautapeli. Pelin ideana, oli että jokaisella pelaajalla on erinväriset lasit, joiden avulla jokainen näkee vain oman väriset putket, ja tarkoituksena liikuttamalla omia putkia saada mahdollisimman pitkä reitti putkista. Henkilö joka saa ensimmäisenä esimerkiksi 20 putken yhtenäisen reitin ensimmäisenä voittoa. Putkien liikuttaminen tapahtuu ottamalla yksi putki pois kentältä ja laittamalla putki takaisin reunalta. Reunalta laitettavan palikan pitää täyttää tyhjäksi jäänyt kohta kentässä. (Oulu Game Lab 2013.)

Idea pääsi Oulu Game Labissa vaiheeseen jossa ideasta ryhdyttiin tekemään demoa. Tähän mennessä idea oli muuttunut lautapelista hamsteri-aiheiseksi Facebook-peliksi. Pelin tarina oli, että pelissä Henry niminen hamsteri haluaa päästä pois lemmikkieläinkaupasta ja pelaaja auttaa sitä pakenemaan. Tarinassa oli myös muita hahmoja kuten Alfred, joka on vanhempi hamsteri ja se auttaa ja ohjaistaa pelaajaa pelin aikana, sekä Amy-pähkinä, joka on Henryn kaverina seikkailussa.

5.2 Oulu Game Lab ja suunnittelu

Oulu Game Lab on Oulun ammattikorkeakoulun 2012 perustama harjoittelu- ja kehitysohjelma, joka tukee pelinkehitystä prototyyppeihin, tuotteisiin ja startup-yrityksiin asti. Oulu Game Lab on räätälöity vastaamaan pelialan tarpeisiin (Oulu Game Lab 2014a, viitattu 9.9.2014). Oulu Game Labista on saatu 28.9.2014 mennessä 9 startup-yritystä, 25 demoa, 90 konsepti-ideaa sekä 170 uutta alan ammattilaista. Oulu Game Lab -koulutusohjelma alkaa kaksi kertaa vuodessa ja auttaa uusia pelialan yrityksiä kasvamaan ja tuottamaan pelialan työpaikkoja Oulun alueelle (Oulu Game Lab 2014b, viitattu 28.9.2014).

Ennen HamsterScapea suunnitelmassa oli tehdä lautapeli jossa kaksi pelaajaa pelaa toisiaan vastaan liikuttelemalla palikoita laudalla. Idea vaihdettiin Oulu Game Labin aikana Facebook-peliin, jossa liikutellaan palikoita samalla periaatteella. Pelin ansaitalogiikka on tehdä Facebookiin kauppa josta voi ostaa peliä helpottavia buustereita, joiden avulla pelaaja voi muuttaa esimerkiksi yhden laudalla olevan palikan toisenlaiseksi ja näin helpottaa kentän läpäisyä.

5.3 Markkinointi

HamsterScapea markkinoitiin noin 30-vuotiaille naisille, jotka pitävät peleistä, jossa on söpöjä ja hassuja hahmoja. Testaus vaiheessa huomattiin että pelistä pitivät eniten 20-30 miehet. Markkinointitapana oli Facebookin kautta laitettu markkinointi, koska siellä voi määritellä kenelle mainos näkyy. Ongelmaksi muodostui Facebook-pelin markkinointi. Unityllä tehty, selaimessa toimiva peli nimittäin tarvitsee toimiakseen Unity Web Player -ohjelman, joka käyttäjän tarvitsee ladata erikseen. Mainosta klikanneet käyttäjät eivät koskaan ladanneet Unity Web Playeria, siksi pelille ei myöskään saatu käyttäjiä.

6 TESTAUS

Aikaisemmin ohjelmistotuotannon testauksessa oli tarkoitus varmistaa ohjelman toimiminen. Nykyään testauksen tarkoitus on varmistaa ohjelman laatu, jotta ohjelmisto täyttää kaikki tarvittavat laatuvaatimukset. Testaajien tarkoitus on löytää nykyään ohjelmistosta niin paljon virheitä kuin mahdollista ja näin ollen ohjelma saadaan tekemään sitä mitä sen pitääkin tehdä. (Sainio 2009, viitattu 9.9.2014.)

6.1 Demo-versio

Demo-version kehitysvaiheessa testaus aloitettiin heti kun oli jotain testattavaa, jotta tiedettiin tarkkaan mitä muutoksia peliin pitää tehdä, jotta pelaaminen helpottuisi. Demo-version kehityksessä toimistolle kutsuttiin pelaajia testaamaan peliä ja pyydettiin palautetta pelistä.

Demo-version testauksissa huomattiin pelaajilla olevan ongelmia ymmärtää pelin logiikkaa, joten silloin päätettiin peliin tehdä tutoriaali. Ensimmäisessä kentässä peli opettaa pelaajaa pelaamaan näyttämällä mitä palikkaa pitää liikuttaa. Näin ensimmäisen kentän pääsee aina läpi.

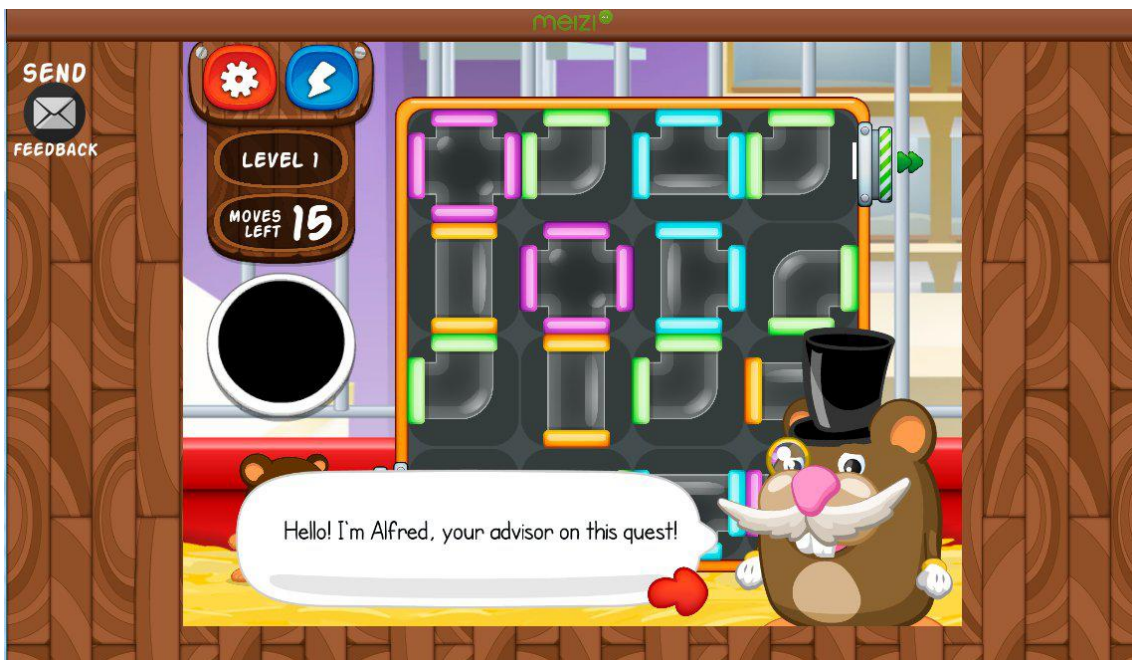
6.2 Julkaisu-versio

Demon jälkeen kehittäjä-tiimistä lähti testauksesta vastuussa olleen henkilö, joten julkaisu-version kehityksessä testaus toteutettiin Facebookin kautta ja kehitettiin sieltä saadun palautteen perusteella. Julkaisua varten peliä muutettiin yksinkertaisemmaksi ja lopulta julkaisuun tehtiin Challenge-versio pelistä, jossa on yksi kenttä. Tämä ratkaisu teki pelaajille ongelmaksi ymmärtää peliä, koska tutoriaaleja pelissä ei enää ollut samalla tavalla.

HamsterScapen Facebook-versiota tehtiin myös Androidille. Sitä ei koskaan julkaistu, koska keskityttiin tekemään Hamsterscape Weekly Challengea. Ensimmäistä kertaa kokeillessamme peliä Android-mobiililaitteilla se ei toiminut kovin hyvin, vaan peli kaatui liian raskaaseen koodiin. Koodia kevennettiin lopulta paljon. Unityssä pystyy tekemään helposti koodia, joka ajetaan jokainen kuva, eli jos peli toimii 30 kuvaa sekunnissa niin koodia ajetaan 30 kertaa sekunnissa. Koodi, joka ajetaan jokainen kuva, on helppo tehdä toimimaan, mutta samalla se tekee helposti koodista todella raskasta. Peliä testattiin Samsung Galaxy S3:lla ja Samsung Galaxy S1 Plussalla.

6.3 Hyödyt

Testaus on pelinkehityksessä todella tärkeää, jotta saadaan varmistettua pelin toimivuus pelaajan näkökulmasta ja minimoitua pelissä olevia virheitä. HamsterScapen kaltaisessa pelissä on tärkeää saada pelaaja ymmärtämään pelin logiikka mahdollisimman nopeasti. Jos pelaaja ei opi peliä tarpeeksi nopeasti, pelaaja yksinkertaisesti lopettaa pelaamisen eikä enää palaa.



KUVIO 4. Ensimmäisen kentän tutoriaali.

Pelien ja ohjelmistojen kehityksessä ohjelmista löytyy aina jonkinlaisia virheitä. Testauksen ansiosta virheiden määrää saadaan tiputettua mahdollisimman alas, jotta julkaisuun saadaan mahdollisimman hyvin toimiva peli. HamsterScapessa virheitä tuli, kun putkessa liikkuva hamsteri jumittui putkeen tai pistetaulukko näkyy väärin. Nämä virheet saatiin korjattua.

7 HAMSTERSCAPE-DEMON KEHITYSTYÖ

HamsterScapea suunniteltiin kaksi kuukautta ennen varsinaista ohjelmointityötä. Tämän ansiosta kaikilla on todella hyvin selvillä, mitä pitää tehdä ja mitä demossa on tarkoitus olla valmiina. Peliä suunniteltiin Oulu Game Labissa osana pelialan koulutusta

7.1 Vaatimukset

Demossa vaatimuksena oli saada kahdeksan kenttää ja peli toimimaan. Haastavimmat asiat oli saada palikoiden liikkuttelu kentässä mahdollisimman toimivaksi, sekä pelin piti osata katsoa, milloin reitti on valmis ja hamsterin piti osata liikkua putkessa ongelmitta. Ongelmia tuotti lisäksi Facebook yhteyden toimiminen. Yhteyttä tarvittiin kavereiden löytämiseen ja tulosten vertailuun. Demossa ei ollut tarkoitus saada kauppaa toimimaan ja se jätettiin suoraan pois, jotta aikaa jäisi tarpeeksi itse pelin toimimisen varmistamiseen.

HamsterScapen demo oli näkyvillä OGL-expossa vuoden 2013 joulukuussa, jossa se äänestettiin parhaaksi esillä olevaksi demoksi. OGL-expo on Oulu Game Labin järjestämä tapahtuma, jossa esitellään demot, joita tekivät Oulu Game Labissa syksyllä aloittaneet henkilöt.

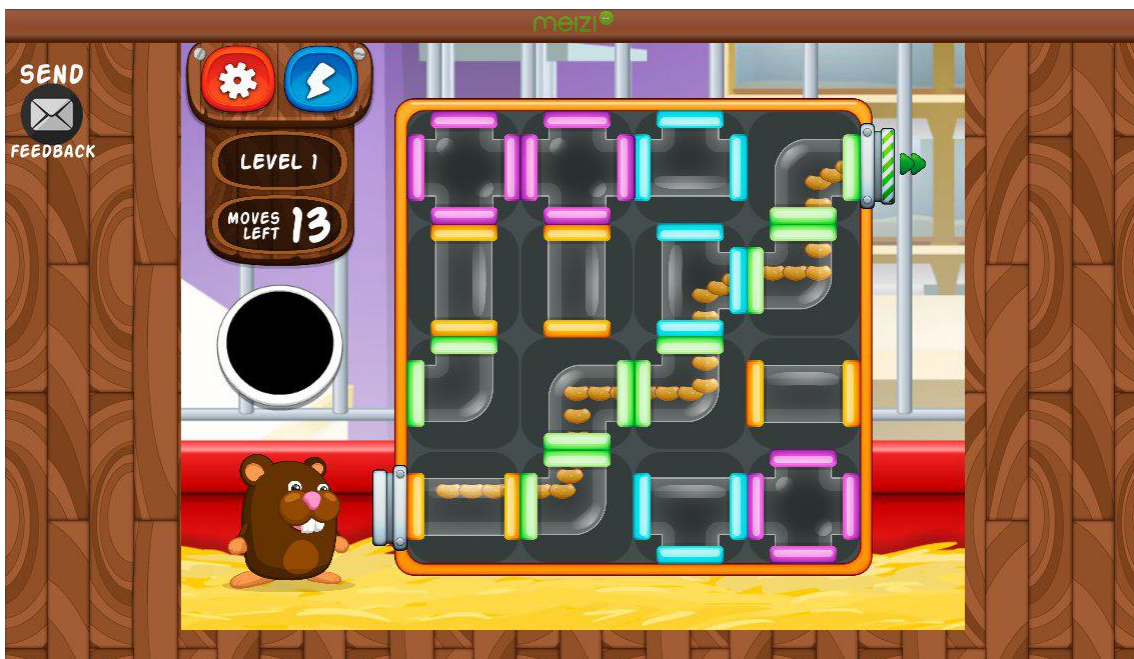
7.2 Aikataulu

Demon tekemisessä ensimmäiset kaksi kuukautta käytettiin pelin suunnitteluun, jotta kaikille olisi demoa aloittaessa varmaa, mitä kunkin pitää tehdä. Tämä suunnitelma onnistui. Kun peliä alettiin tehdä, oli kaikille todella hyvin tiedossa, mitä kukakin teki ja työ oli tehokasta.

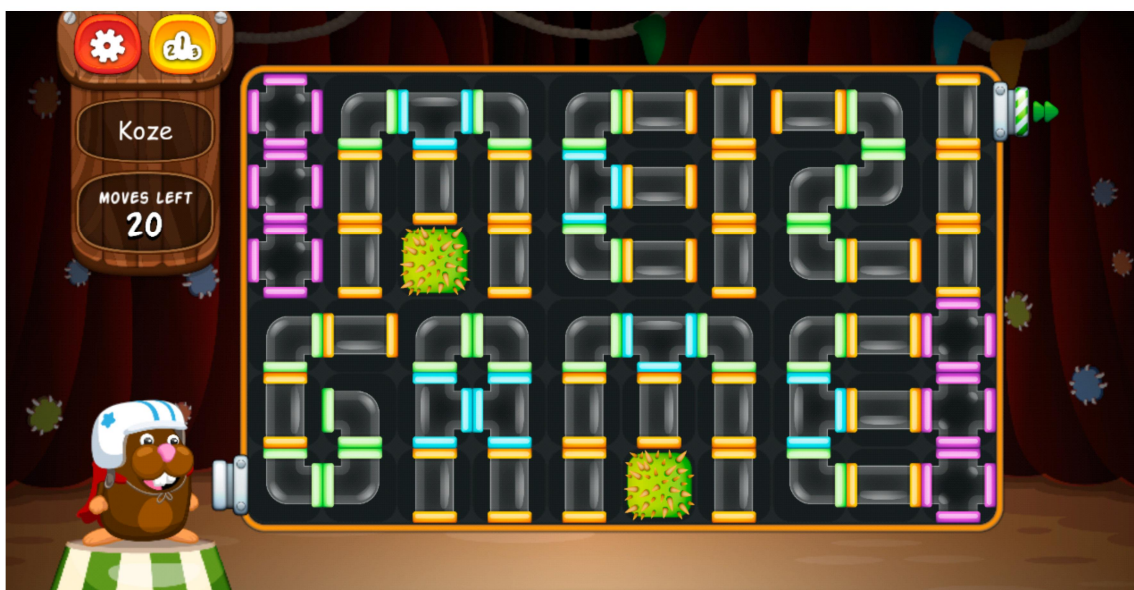
Ohjelmoinnissa ongelmia tuotti alussa se, että Unity oli uusi ohjelma ohjelmoijille ja alussa ohjelman opetteluun meni aikaa. Siitä huolimatta pysyttiin aikataulussa. Projektia varten tehtiin selvä aikataulu, jotta tiedettiin mitä milläkin viikolla oli tarkoitus olla valmista ja mitkä ominaisuudet piti olla lopullisessa demossa valmiina.

8 HAMSTERSCAPE-JULKAISUVERSION KEHITYSTYÖ

HamsterScapen vaatimukset muuttuivat vähän kesken projektin ja muiden projektien alkaminen samaanaikaan hidasti pelin kehitystä. Rahoittajan saaminen oli myös vaikeaa aloittelevalla yritykselle. Tässä luvussa kerrotaan, mitä kehityksen aikana tapahtui ja miksi.



KUVIO 5. Alkuperäisen Facebook-pelin ulkonäkö.



Kuvio 6. Androidiin julkaistu peli

8.1 Vaatimukset

Alun perin HamsterScapen suunnitelmissa oli saada julkaisuun noin 100 kenttää, mutta myöhemmin päätettiin tehdä pienempi peli, jota voitaisiin näyttää rahoittajille. Pienempi peli julkaistiin Androidille, jossa on viikottain vaihtuva kenttä ja tämä kenttä on vaikeampi kuin normaalikentät. Pienemmän pelin nimi on Hamsterscape Weekly Challenge.

Kun projekteja tuli lisää jäi alkuperäisen HamsterScapen kehitys vähemmälle ja vaatimuksia alettiin muuttamaan, jolloin tarkoituksena oli saada 30 kenttää lopulliseen julkaisuun. Rahoituksen saaminen aloittavalle yritykselle on vaikeaa, kun yrityksellä ei ole näytettävää osaamisesta. Siksi yrityksessä päätettiin tehdä pienempiä pelejä, jotta saataisiin pelaajia peleihin ja näin ollen rahoittajat kiinnostumaan. HamsterScapen kehitykseen haetaan rahoitusta, jotta projektia päästään jatkamaan.

8.2 Ohjelmointi

Pelin ohjelmointi tehtiin C#-kielellä. Pyrittiin että käytetään mahdollisimman vähän Unityn Asset Storen asset-tiedostoja, mikäli jos ohjelmoinnissa tulee jotain ongelmia, niin osataan korjata ne mahdollisimman helposti. Alkuperäisen HamsterScapen koodin pohjalla pystyi muokkaamaan peliä niin, että siinä oli yksi kenttä, joka oli isompi.

Peliin tehtiin mahdollisuus yhdistää se Facebookiin, tällöin pelaaja saa listan kavereiden tekemistä tuloksista ja pystyi näin ollen kilpailemaan kavereita vastaan. Peli julkaistiin Androidille, jonka takia siihen täytyi ohjelmoida asioita mitä suoritetaan, riippuen millä alustalla peliä pelataan. Valikoiden toimintaan tehtiin toisen kameran avulla. Kun toiseen kameraan tulee valikko esiin, ottaa peli sen kameran klikkauksen huomioon ja silloin pelaaja pystyi liikkumaan valikoissa. Jos pelaaja lähtee pois, valikot piilotetaan ja kameraksi vaihdetaan alkuperäinen kamera.

HamsterScapessa käytettiin asseteina Facebook UnitySDK:ta facebookiin yhdistämiseen, Astaria tarkistukseen onko reitti valmis ja iTweeniä, jotta pelissä liikuteltavien palikoiden liikuttaminen näyttäisi paremmalta. Asset-tiedostojen käytössä oli kuitenkin tietyt riskit. Kun asset-tiedostoa alkaa käyttämään, tulee peliin koodia, jota ei ole itse tehnyt. Jos ongelmia ilmenee, virheen korjaamiseen saattaa mennä paljon aikaa. Lisäksi pelin optimoinnin kannalta on parempi, että asetteja ei käytetä, koska usein assetit on tehty tekemään paljon muutakin kuin sen asian mihin sitä käytetään ja projektiin tulee paljon ylimääräistä koodia. Asetteja joutuu usein myös muokkaamaan hieman juuri itselle sopiviksi. Kun asset-tiedostoon tulee päivitys, voi päivitys rikkoa koko projektin ja sen korjaaminen saattaa viedä paljon aikaa.

9 POHDINTA

Opinnäytetyössäni jouduin käyttämään suuria tuntimääriä ohjelmointia ennen kuin aloitin tekemään teoriaosuutta. Tietoa oli internetissä todella paljon ja tiedon löytäminen ei tuottanut suurta ongelmaa.

Projektin demovaiheen jälkeen ollut koodi ei ollut parasta mahdollista ja skriptit oli nimetty vaikeasti, mikä hankaloitti jatkokehitystä. Tämä asia ei paljoa haitannut, koska olin itse tehnyt koodin. Tuohon asiaan pitäisi keskittyä enemmän, koska jos tiimiin tulee uusia ohjelmoijia heillä olisi todella vaikea alkaa tekemään projektia. Paljon aikaa menisi hukkaan vain koodin läpikäymiseen.

Työssä oli tarkoitus alun perin tehdä Facebook-peli Hamsterscape. Jotta peli saataisiin nopeammin julkaistua ja näyttöä siitä, mitä osataan, päätettiin pelistä julkaista pienempi versio Hamsterscape Weekly Challenge. Hamsterscapea tehtiin muutamia kenttiä eteenpäin, mutta tulevaisuus näyttää, tullaanko sitä lopulta tekemään loppuun.

Hamsterscape Weekly Challenge valmistui hyvin ja se saatiin vietyä onnistuneesti mobiililaitteille, vaikka alussa olikin ongelmia, vanhemmilla laitteilla. Peli julkaistiin lopulta Google Play -kaupassa 22. päivä heinäkuuta 2014.

10 LÄHTEET

Catone J. 2009, HOW TO: Find and Add Facebook Apps, viitattu 10.8.2014.
<http://mashable.com/2009/10/24/facebook-apps-guide/>

Catlike Coding 2014, Unity C# Tutorials, viitattu 16.8.2014.
<http://catlikecoding.com/unity/tutorials/>

Glasses, Oulu Game Lab. 2013. Projektin Glasses alkuperäissuunnitelma. (Ei julkaisupaikkaa tai julkaisijaa).

Kajaanin ammattikorkeakoulun kirjasto 2014. Tiedon hakijan ja löytäjän opas. viitattu 10.8.2014, <http://gallia.kajak.fi/opetusyhteistyö/kirjasto/Facebook.htm>

NetMarketShare 2014, Mobile/Tablet Operating System Market Share, viitattu 29.9.2014, <http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomid=1>

Oulu Game Lab 2014a, Oulu Game Lab, viitattu 9.9.2014.
<http://www.oulugamelab.net/>

Oulu Game Lab 2014b, Overview — Oulu Game Lab, viitattu 28.9.2014.
<http://www.oulugamelab.net/overview/>

Sainio L. 2009, OHJELMISTOTESTAUKSEN MENETELMÄT JA TYÖVÄLINEET, viitattu 9.9.2014,
https://www.theseus.fi/bitstream/handle/10024/12297/Sainio_Laura_liite1.pdf

The Codehaus 2006, Duck Typing, viitattu 16.8.2014.
<http://boo.codehaus.org/Duck+Typing>

Unity Technologies 2014a, What is Unity?, viitattu 10.8.2014.
<http://unity3d.com/pages/what-is-unity>

Unity Technologies 2014b, Fast Facts, viitattu 10.8.2014.
<https://unity3d.com/public-relations>

Unity Technologies 2014c, Unity scripting, viitattu 10.8.2014.
<https://unity3d.com/unity/workflow/scripting>

Unity Technologies 2014d, C# vs JS syntax, viitattu 16.8.2014.
<http://unity3d.com/learn/tutorials/modules/beginner/scripting/c-sharp-vs-javascript-syntax>

Unity Technologies 2014e, Unity - Manual: Asset Store, viitattu 13.9.2014.
<http://docs.unity3d.com/Manual/AssetStore.html>

Unity Technologies 2014f, Unity - Manual: Asset Store Access and Navigation, viitattu 13.9.2014. <http://docs.unity3d.com/Manual/AccessNavigation.html>

wiki.unity3d.com 2014, UnityScript versus JavaScript, viitattu 13.9.2014,
http://wiki.unity3d.com/index.php?title=UnityScript_versus_JavaScript

Wikimedia Foundation, Inc 2014a, C Sharp (programming language), viitattu 17.9.2014, [http://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](http://en.wikipedia.org/wiki/C_Sharp_(programming_language))

Wikimedia Foundation, Inc 2014b, Android (operating system), viitattu 29.9.2014, [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))

Wikimedia Foundation, Inc 2014c, ios, viitattu 2.10.2014,
<http://fi.wikipedia.org/wiki/IOS>

Wikimedia Foundation, Inc 2014d, Windows Phone, viitattu 2.10.2014,
http://fi.wikipedia.org/wiki/Windows_Phone

Will Pleasant-Ryan 2012, Game Programming in Boo with the Unity Engine,
Part 1: Not All Who Wander Are Lost, viitattu 16.8.2014.
[http://spin.atomicobject.com/2012/12/29/game-programming-boo-unity-
engine-part-1/](http://spin.atomicobject.com/2012/12/29/game-programming-boo-unity-engine-part-1/)